Bridging clinical information systems and grid middleware: a Medical Data Manager

Johan Montagnat[1], Daniel Jouvenot[2], Christophe Pera[3], Ákos Frohner[4], Peter Kunszt[4], Birger Koblitz[4], Nuno Santos[4], Cal Loomis[2]

[1] CNRS, I3S laboratory, http://www.i3s.unice.fr/~johan

[2] CNRS, LAL laboratory, http://www.lal.in2p3.fr

[3] CNRS, CREATIS laboratory, http://www.creatis.insa-lyon.fr

[4] CERN, http://www.cern.ch

Abstract

This paper describes the effort to deploy a Medical Data Management service on top of the EGEE grid infrastructure. The most widely accepted medical image standard, DICOM, was developed for fulfilling clinical practice. It is implemented in most medical image acquisition and analysis devices. The EGEE middleware is using the SRM standard for handling grid files. Our prototype is exposing an SRM compliant interface to the grid middleware, transforming on the fly SRM requests into DICOM transactions. The prototype ensures user identification, strict file access control and data protection through the use of relevant grid services. This Medical Data Manager is easing the access to medical databases needed for many medical data analysis applications deployed today. It offers a high level data management service, compatible with clinical practices, which encourages the migration of medical applications towards grid infrastructures. A limited scale testbed has been deployed as a proof of concept of this new service. The service is expected to be put into production with the next EGEE middleware generation.

1 Medical data management in hospitals and grid data management

The medical community is routinely using clinical images and associated medical data for diagnosis, intervention planning and therapy follow-up. Medical imagers are producing an increasing number of digital images for which computerized archiving, processing and analysis are needed [8, 12]. Indeed, image networks have become a critical component of the daily clinical practice over the years. With their emergence, the need for standardized medical data formats and exchange procedures has grown [2]. For this reason, the *Digital Image and COmmunication in Medicine* standard (DICOM) [6] was adopted by a large consortium of medical device vendors. *Picture Archiving and Communication Systems* (PACS) [10], manipulating DICOM images and often other medical data in proprietary formats, are proposed by medical device vendors for managing clinical data. PACS are often proprietary solutions weakly standardized. PACS may be more or less connected to the *Hospital Information System* (HIS), holding administrative information about patients, and *Radiological Information Systems* (RIS), holding additional information for the radiological departments. The DICOM standard, PACS, RIS and HIS have been

developed with clinical needs in mind. They are easing the daily care of the patients and medical administrative procedures. However, their usage in other areas is very limited. The interface with computing infrastructures for instance is almost completely lacking. In addition, current PACS hardly address medical data management needs beyond clinical centers' administrative boundaries, while the patient medical folders are often wide spread over many medical sites that have been involved in the patient's healthcare. Many medical image acquisition devices are also weakly conforming to the DICOM standard, thus hardly hiding the heterogeneity of these systems.

In the last decades, with the growing availability of digital medical data, many medical data processing and analysis algorithms were developed, enabling computerized medical applications for the benefit of the patient and healthcare practitioners. Although sharing the same data sources, the medical image analysis community has different requirements for medical system than the healthcare community. Many algorithms are developed for processing and producing image files. A common procedure for accessing all medical data sources is needed.

Given the enormous amount of medical data produced inside hospitals and the cost of medical data computing (especially image analysis algorithms), grid proved to be very useful infrastructures for a large variety of medical applications [11]. Grids are providing computing resources and workload systems that ease application code deployment and usage. Moreover, grids are providing distributed data management services that are well suited for handling medical data geographically spread throughout various medical centers [5, 7, 4, 9, 3]. However, existing grid middlewares are often only dealing with data files and do not provide higher level services for manipulating medical data. Medical data often have to be manually transferred and transformed from hospital sources to grid storage before being processed and analyzed. Such manual interventions are tedious and often limit systematic use of grid infrastructures. In some cases, they may even prevent the use of grids, *e.g.* when the amount of data to transfer is too large. As a consequence, the first key to the success of the systematic deployment of medical image processing algorithms is to provide a data manager that:

- Provides access to medical data sources for computing without interfering with the clinical practice.
- Ensures transparency so that accessing medical data does not require any specific user intervention.
- Ensures a high data protection level to respect patients privacy.

The Medical Data Manager (MDM) service described in this paper was designed to fulfill these constraints. It was developed with the support of the EGEE[1] European IST project. The remaining of this paper describes the technical requirements to be addressed for such a service and details the service design.

2 Clinical usage of medical data

The DICOM standard introduced earlier encompasses, among other things, an image format and an image communication protocol. A DICOM image usually contains one slice (a 2D image) acquired using any medical imaging modality (MRI, CT-scan, PET, SPECT, ultrasound, X-ray... [1]). A DICOM image may contain a multi-slice data set but this is rarely encountered. A DICOM image contains both the image data itself and a set of additional information (or *metadata*) related to

[1] Enabling Grids for E-sciencE, http://www.eu-egee.org

the image, the patient, the acquisition parameters and the radiology department. DICOM metadata are stored in fields. Each field is identified by a unique tag defined in the DICOM standard. A given field may be present or absent depending on the imager that produced the image. The standard is open and image device manufacturers tend to use their own fields for various information. A couple of fields (such as image size) are mandatory but experience proved that surprises should be expected when analyzing a DICOM image. The image itself is usually stored as raw data. Most imaging devices produce one intensity value per image pixel, coded in a 12 bit format. Other format may be encountered such as 16 bit data or lossless JPEG.

2.1 DICOM protocol, storage, and security

Most (reasonably modern) medical image acquisition device are DICOM clients. DICOM servers are computers with on-disk and/or tape back-ends able to store and retrieve DICOM images. The DICOM protocol defines the communication protocol between DICOM servers and clients.

There is no standardization on DICOM storage. DICOM servers are implementing their own policy of data storage. One should not see DICOM data sets as a set of files. As stated above, a single DICOM image usually contains only one image slice. In practice, during a medical examination (a DICOM *study*), a radiologist acquires several 2D and 3D images, representing up to hundreds to thousands of slices. A study is divided in one or several *series* and each serie is composed by a set of slices (that can be stacked to assemble a a volume when they belong to the same 3D image). Note that there is often no notion of 3D image encoded in the DICOM format: a serie may contain a set of slices composing several 3D images. The way a DICOM server stores these data sets on disk is irrelevant just like the way a database stores its table is usually not known from the users: the medical user is never exposed to the DICOM storage and does not need to know if different files are used for each DICOM slice, serie, study, etc. Metadata are included in DICOM image headers, making them difficult to manipulate. A DICOM server will often extract these metadata and store them in a database to ease data search.

The DICOM security model is rather weak. DICOM files are unencrypted and transported unencrypted. Files contain patient data. The DICOM server security model is based on a per-application basis: all users having access to some DICOM client application can access to the information that the server returns to this specific application. DICOM servers are using random file names without any connection to the patient information and a proprietary data storage policy. To cope with these data protection limitations, security is often implemented in hospitals by isolating the images network from the outside world.

2.2 Access to medical images

Each image acquisition device is a potential DICOM compliant medical image source. In a radiological department, one or several DICOM servers can be set up to centralize data acquired on this site. Medical data are naturally distributed over the different acquisition sites.

In clinical practice, physicians do not access directly to image files. They identify data by associated metadata such as patient name, acquisition date, radiologist name, etc. The data are transferred mainly for visualization purposes. The physician quickly scans the slices stack in the DICOM study and focuses on the slices he or she is interested in.

3 Medical image analysis

In the medical image analysis community, the needs are quite different. One often needs to identify images through metadata too, although the search are not necessarily for nominative data but often related to the acquisition type or body region. 3D images are exported to disk files for post-processing and ease of use. Various 3D medical image format may be used to stack different DICOM slices into a single image volume (the most common being the *analyze* file format).

3.1 Enforcing medical data and security

All medical data should be considered as sensitive to preserve patient privacy. Nominative medical data are of course the most critical data and therefore, no binding between nominative data and images should be possible for non accredited users. In clinical practice, this result is often obtained by isolation of the image network. Only physicians participating to one patient healthcare should have access to the data of this patient.

On a grid, the distribution of data make security problem very sensitive. To ensure patient privacy, the header of all DICOM images sent by a DICOM server should be wiped out, at least partially, to ensure anonymity. All images that are stored out of the source center should be encrypted to ensure that non accredited users cannot read the image content.

4 Medical Data Management Service

4.1 EGEE grid middleware

The EGEE project is currently deploying the LCG2 middleware[2] on its production infrastructure. LCG2 is based on GLOBUS2, Condor, and the other services developed in the European DataGrid project[3]. A new generation middleware, gLite[4], is under testing and should be deployed at Spring 2006. Our Medical Data Manager service (MDM) is based on gLite.

The gLite middleware provides workload management services for submitting computing tasks to the grid infrastructure and data management services for managing distributed files. The data management is based on a set of *Storage Elements* which are storage resources distributed in the various sites participating in the infrastructure (currently, more than 180 sites distributed all over Europe and beyond). All storage elements expose a same interface for interacting with the other middleware services: the *Storage Resource Manager* interface (SRM) that is standardized in the context of the Global Grid Forum[5]. The SRM is handling local data at a file level. It offers an interface to create, fetch, pin, or destroy files among other things. It does not implement data transfer by itself. Additional services such as GridFTP or gLiteIO are coexisting on storage elements to provide transfer capabilities.

In addition to storage resources, the gLite data management system includes a *File Catalog* (Fireman) offering a unique entry point for files distributed on all grid storage elements. Each file is uniquely identified through a *Global Unique IDentifier* (GUID). The file catalog contains tables associating each GUID to file location. For efficiency and fault tolerance reasons, files may be replicated on different sites. Thus, each GUID may be associated to several locations. To ease the manipulation by

[2] LCG2: Large hadron collider Computing Grid middleware, http://lcg-web.cern.ch
[3] European DataGrid project, http://www.edg.org
[4] gLite middleware, http://www.glite.org
[5] Global Grid Forum, http://www.ggf.org

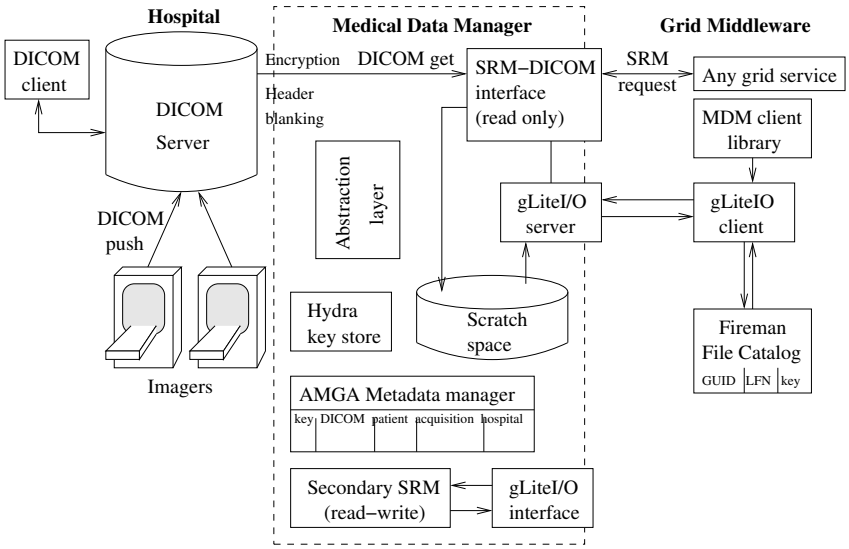

Figure 1: Overview of the medical data manager

users, human readable *Logical File Names* (LFN) can be associated to each file (each GUID).

4.2 Medical Data Management service design

The Medical Data Management service architecture is diagrammed in figure 1. On the left, is represented a clinical site: various imagers in an hospital are *pushing* the images produced on a DICOM server. Inside the hospital, clinicians can access the DICOM server content through DICOM clients. In the center of figure 1, the MDM internal logic is represented. On the right side, the grid services interfacing with the MDM are shown.

All middleware services requiring access to data storage do so through SRM requests sent to storage elements. To remain compatible with the rest of the grid infrastructure, our MDM service is based on a SRM-DICOM interface software. The SRM-DICOM core is receiving SRM requests and transforms them into DICOM transactions addressed to the medical servers. Thus, medical data servers can be shared between clinicians (using the classical DICOM interface inside hospitals) and image analysis scientists (using the SRM-DICOM interface to access the same data bases) without interfering with the clinical practice. An internal scratch space is used to transform DICOM data into files that are accessible through data transfer services (GridFTP or gLiteIO).

A metadata manager is also used to extract DICOM headers information and ease data search. The AMGA[6] service [13] is used for ensuring secured storage of these very sensitive data. The AMGA server holds a relation between each DICOM slice and the image metadata.

This specialized SRM is not providing a classical Read/Write interface to a storage element. A classical R/W storage element can symmetrically receive grid files to be stored or deliver archived files to the grid on request. In the MDM, The SRM interface only accepts registration request coming internally from the hospital. To avoid interfering with the clinical data, external grid files are not permitted to

[6]ARDA metadata catalog project, http://project-arda-dev.web.cern.ch/project-arda-dev/metadata/

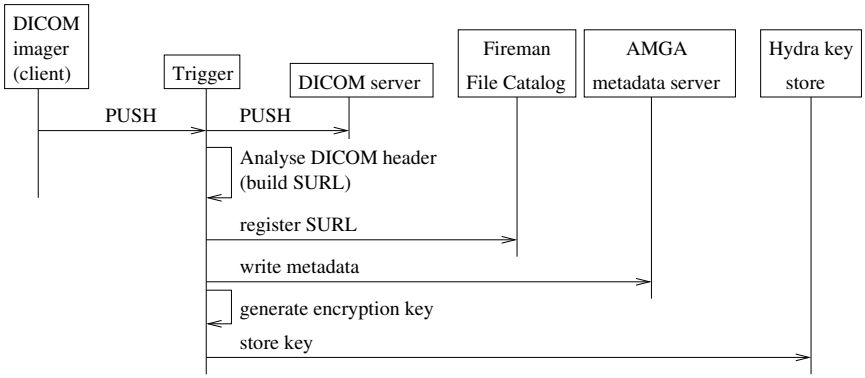

Figure 2: Triggered action at image creation

be registered on the MDM storage space: only get requests are authorized from the grid side. If classical grid storage is desired (with write capability), a classical secondary SRM can be installed on the same host.

For data encryption needs, a secured encryption key catalog is also used. It is named *hydra catalog* as it uses a split key storage strategy to improve security and fault tolerance [15, 14].

An *Abstraction layer*, currently being prototyped and tested, is also depicted on the diagram. Its role is to offer a higher level abstraction for accessing 3D images by associating all DICOM slices corresponding to a single volume. Indeed, most medical image processing applications are not manipulating 2D images independently but rather consider complete volumes. The abstraction layer is associating a single GUID to each volume. On a request for the volume associated to this GUID, all corresponding slices are transferred from the DICOM server and assembled in a single volume in scratch space.

4.3 Internal service interaction patterns

To fulfill its role, the MDM service needs to be notified when files are produced by the imagers and stored into the DICOM server. This notification triggers a file registration procedure that is depicted in figure 2. The DICOM data triggering the operation is first stored into the hospital DICOM server as usual. The DICOM header is then analyzed to extract image identifying information. This DICOM ID is used to build a *Storage URL* (SURL) as used by the grid File Catalog to locate files. The SURL is registered into the File Catalog and a GUID associated to this data on the grid side. The other metadata extracted from the DICOM header are stored into the AMGA metadata server. Finally, encryption keys that are associated to the file and that will be used for data retrieval are stored into the hydra distributed database.

Once DICOM data sets have been registered into the MDM, the server is able to deliver requested data to the grid as depicted in figure 3. A client library is used for this purpose. To cover all application use cases, the MDM client library provides APIs for requesting files based on their grid identifier (GUID) or the metadata attached to the file. In case of request on the metadata, a database query is first made to the AMGA server and the list of GUIDs of images matching the query are returned. The SRM-DICOM server can then deliver images requested through their GUID. SRM get requests are translated into DICOM get queries. Data extracted from the DICOM server are first written to an internal scratch space. Their format is transformed into a simple 3D image file format (a human readable header including

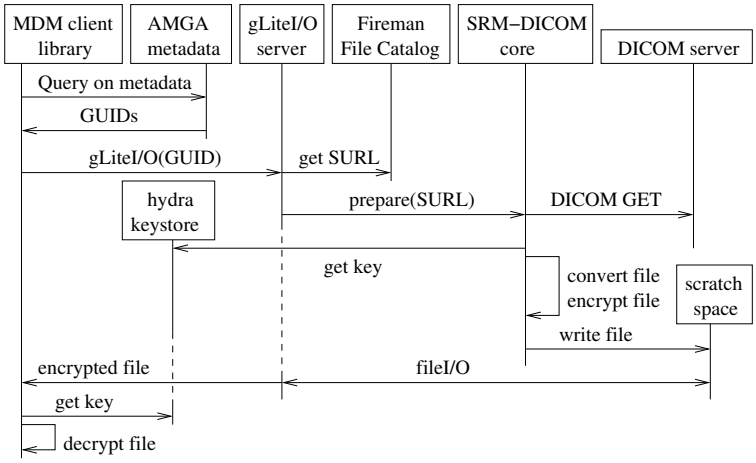

Figure 3: Accessing DICOM images

image size and encoding, followed by the raw image data). In this transformation, the DICOM header, containing patient identifying operations, are lost to preserve anonymity. The files are also encrypted before being sent out to ensure that no sensitive information is never transfered nor stored on the grid in a readable format. Files are then transferred through the gLiteIO service and returned to the client in an encrypted form. The file is only decrypted in memory of the client host, given that the client is authorized to access the file encryption keys.

4.4 MDM client

On the client side, three levels of interfaces are available to access and manipulate the data hold by the MDM. The MDM is seen from the middleware as any storage resource exposing a standard SRM interface, the standard data management client interface can be used to access images provided that their GUID is known. The files retrieved using this standard interface are encrypted. The second interface is an extra middleware layer which encompasses access to the encryption key and the SRM. Thus images can be fetched and decrypted locally. The third and last level of interface is the fully MDM aware client library represented in figure 3. It provides access to encrypted files and in-memory decryption of the data on the application side, plus access to the metadata through the AMGA client interface.

5 Discussion

5.1 Data security

The security model of the MDM relies on several services: (i) file access control, (ii) files anonymization, (iii) files encryption, and (iv) secured access to metadata. The user is coherently identified through a single X509 certificate and all services involved in security are using the same identification procedure. The file access control is enforced by the gLiteIO service which accepts Access Control Lists (ACLs) for fine grained access control. The hydra key store and the AMGA metadata services also accept ACLs. To read an image content, a user needs to be authorized both to access the file and to the encryption key. The access rights to the sensitive metadata associated to the files are administrated independently. Thus, it is possible to grant access to an encrypted file only (*e.g.* for replicating a file without accessing

to the content), to the file content (*e.g.* for processing the data without revealing the patient identity), or to the full file metadata (*e.g.* for medical usage).

Through ACLs, it is possible to implement complex use cases, granting access rights (for listing, reading, or writing) to patients, physicians, healthcare practitioners, or researchers needing to process medical data, independently from each other.

5.2 Medical metadata schema

A minimal metadata schema is defined in the MDM service for all images stored. It provides basic information on the patient owning the image, the image properties, acquisition parameters, etc. There are two main indexes used: a patient ID, for all nominative information associated to patients and the image GUID for all information associated to images. The patient ID is a unique but irreversible field (such as a MD5 sum on the patient field name). Four main relational tables are used:

- The Patient table, indexed on the patient ID, contains the most sensitive identifying data (patient name, sex, date of birth, etc).
- The Image table, indexed on the image GUID, contains technical information about the image (size, encoding, etc). It establishes a relation with the patient ID.
- The Medical table, indexed on the image GUID, contains additional information on the acquisition (image modality, acquisition place and date, radiologists, etc).
- The DICOM table, indexed on the image GUID, contains the image DICOM identifiers used for querying the DICOM server.

To remain extensible, an additional Protocol table associates image GUIDs with medical protocol name. Through AMGA, the user can create as many medical protocols as needed, containing specific information related to some particular acquisition (*e.g.* a temporal protocol for cardiac acquisitions, etc). AMGA also enables per table access right control, allowing restricting access to the most sensitive data (*e.g.* the Patient table) to the minimum number of users.

6 Testbed

The Medical Data Manager has been deployed on several sites for testing purposes. Three sites are actually holding data in three DICOM servers installed at I3S (Sophia Antipolis, France), LAL (Orsay, France) and CREATIS (Lyon, France). In addition to the DICOM servers, these sites have installed the core MDM services: a SRM-DICOM server and associated database back-end, a gLiteIO service, a GridFTP service, and all dependencies in the gLite middleware. Client have been deployed on all these three sites.

To complete the installation, an AMGA catalog has also been set up in CREATIS (Lyon) for holding all sites' metadata, and an hydra key store is deployed at CERN (Geneva, Switzerland) for keeping file encryption keys.

Given the number of services involved, the installation and configuration procedure is currently complex. It is being worked out to ease the testbed extension. The MDM service should be deployed in hospitals where little support is provided for the informatics infrastructure.

The testbed deployed has been used to demonstrate the viability of the service by registering and retrieving DICOM files across sites. For testing purposes, DICOM data registrations are triggered by hand. Registered files could be retrieved

and used from EGEE grid nodes transparently, using the standard EGEE data management interface. The next important milestone will be to experiment the system in connection with hospitals by registering real clinical data freshly acquired and registered on the fly from the hospital imagers. This step involves entering a more complex clinical protocol with strong guarantee on the data privacy protection. The security cannot be neglected at any level at this point.

7 Conclusion and future work

The Medical Data Manager service presented in this paper is an important milestone for enabling medical image processing applications on a grid infrastructure. Its main strength are:

- To access medical databases without interfering with clinical practice. Data are kept on clinical sites and transparently transferred to the grid only when needed.
- To expose standard interfaces to other grid services. The MDM is fully integrated in the gLite middleware.
- To ensure a high level of security to preserve patients privacy.

The MDM prototype was successfully deployed and tested in a controlled computing environment. The next step will see interfacing to medical imagers inside hospitals. It will require to simplify the installation and configuration procedures as most as possible.

The core MDM development is not finished yet and additional functionalities will be included to enrich the service. In particular, the abstraction layer depicted in figure 1 will soon be available. Applications will then be able to retrieve 3D volume files rather than single slices. In addition, metadata are expected to be distributed in the different clinical sites where data are acquired rather than being centralized as it is the case in our testbed. This configuration will be more acceptable to the clinical world to keep control on the hospital data. It will require deploying several AMGA servers on different sites and exposing a centralized query service able to retrieve data from these different servers.

Acknowledgments

We are grateful to the EGEE European IST project for providing resources and support to this service development.

References

[1] R. Acharya, R. Wasserman, J. Sevens, and C. Hinojosa. Biomedical Imaging Modalities: a Tutorial. *Computerized Medical Imaging and Graphics*, 19(1):3–25, 1995.

[2] K.P. Andriole, R.L. Morin, Arenson; R.L., J.A. Carrino, B.J. Erickson, S.C. Horii, D.W. Piraino, B.I. Reiner, J.A. Seibert, and E. Siegel. Addressing the Coming Radiology Crisis: The Society for Computer Applications in Radiology SCAR Transforming the Radiological Interpretation Process (TRIP) initiative. *Journal of Digital Imaging*, 17(4):235–243, December 2004.

[3] C. Barillot, R. Valabregue, J.P. Matsumoto, F. Aubry, H. Benali, Y. Cointepas, O. Dameron, M. Dojat, E. Duchesnay, B. Gibaud, S. Kinkingnéhun, D. Papadopoulos, M. Pélégrini-Issac, and E. Simon. NeuroBase: Management of Distributed and Heterogeneous Information Sources inNeuroimaging. In *Distributed Database and processing in Medical Image Computing workshop (DiDaMIC'04)*, Saint Malo, France, September 2004.

[4] I. Blanquer Espert, V. Hernández García, and J.D. Segrelles Quilis. Creating Virtual Storages and Searching DICOM Medical Images through a GRID Middleware based in OGSA. *Journal of Clinical Monitoring and Computing*, 19(4-5):295–305, October 2005.

[5] D. Budgen, M. Turner, I. Kotsiopoulos, F. Zhu, K. Bennett, P. Brereton, J. Keane, P. Layzell, M. Russell, and M. Rigby. Managing healthcare information: the role of the broker. In *Healthgrid'05*, Oxford, UK, April 2005.

[6] DICOM: Digital Imaging and COmmunications in Medicine. http://medical.nema.org/.

[7] M.H. Ellisman, C. Baru, J.S. Grethe, A. Gupta, M. James, B. Ludaescher, M.E. Martone, P.M. Papadopoulos, S.T. Peltier, A. Rajasekar, S. Santini, and I.N. Zaslavsky. Biomedical Informatics Research Network: An Overview. In *Healthgrid'05*, Oxford, UK, April 2005.

[8] C. GERMAIN, V. BRETON, P. CLARYSSE, Y. GAUDEAU, T. GLATARD, E. JEANNOT, Y. LEGRE, C. LOOMIS, I. E. MAGNIN, J. MONTAGNAT, J.-M. Moureaux, A. OSORIO, X. PENNEC, and R. TEXIER. Grid-enabling medical image analysis. *Journal of Clinical Monitoring and Computing*, 19(4-5):339–349, October 2005.

[9] S. Hastings, S. Oster, S. Langella, T.M. Kurc, T. Pan, U.V. Catalyurek, and Saltz J.H. A Grid-based image archival and analysis system. *Journal of the American Medical Informatics Association*, 12:286–295, January 2005.

[10] H. K. Huang. *PACS: Picture Archiving and Communication Systems in Biomedical Imaging*. Hardcover, 1996.

[11] J. Montagnat, F. Bellet, H. Benoit-Cattin, V. Breton, L. Brunie, H. Duque, Y. Legré, I.E. Magnin, L. Maigne, S. Miguet, J.-M. Pierson, L. Seitz, and T. Tweed. Medical images simulation, storage, and processing on the european datagrid testbed. *Journal of Grid Computing*, 2(4):387–400, December 2004.

[12] J. Montagnat, V. Breton, and I.E. Magnin. Using grid technologies to face medical image analysis challenges. In *Biogrid'03, proceedings of the IEEE CCGrid03*, Tokyo, Japan, May 2003.

[13] N. Santos and B. Koblitz. Metadata services on the grid. In *Advanced Computing and Analysis Techniques*, Berlin, Germany, May 2005.

[14] L. Seitz, J.M. Pierson, and L. Brunie. Key management for encrypted data storage in distributed systems. In *IEEE Security in Storage Workshop (SISW)*, Washington DC, USA, October 2003.

[15] L. Seitz, J.M. Pierson, and L. Brunie. Encrypted storage of medical data on a grid. *Methods of Information in Medicine*, 44(2), 2005.